Amendments to the Claims

1. (Currently Amended) A client-server computer system comprising:

a client application server that utilizes data in a particular form and generates a validation request for validation of the data and wherein the request includes the data in an initial form;

an application server accessible by a plurality of client application servers via a plurality of application software protocols, wherein said application server provides a data validation service on the data received from the client application server in response to receiving the validation request from the client application server, wherein the data validation service compares the data in the initial form to validation parameters stored in a hierarchical, table-based system of a plurality of rules organized into at least a FIELD view, a CLASS view and a GLOBAL view, wherein the FIELD view is the view of first priority and contains validation parameters applicable to a specific field, the CLASS view is the view of second priority and contains validation parameters applicable to a class of fields and the GLOBAL view is the default view when there is no match to the data in the FIELD and CLASS views, wherein further the rules included in each of the FIELD view, CLASS view and GLOBAL view are prioritized in a predetermined execution sequence ~~a reference for the particular form utilized by the client application to determine whether the initial form matches the particular form~~ and returns to the same client application server that generated the validation request an indication of valid or invalid, wherein further the data in the initial form is determined to be in a valid format when a ~~date~~ data item specified in the data in the initial form has a non-empty table field, [[a]] the ~~date~~ data item specified in the data in the initial form is in a valid ~~date~~ data format, and when the ~~date~~ data item falls within a set of range of validation values ~~dates~~; and

a storage mass coupled to said application server for storing a system of dynamically maintainable validation functions for performing said validation service.

2. (Previously Presented) A client-server computer system according to claim 1, wherein said storage mass comprises a database.

3. (Original) A client-server computer system according to claim 1, wherein said

validation functions are represented by a storage schema in the form of Lightweight Directory Access Protocol.

4. (Previously Presented) A client-server computer system according to claim 2, wherein said database contains a table-based system of rules organized into at least three hierarchically-organized views.

5. (Original) A client-server computer system according to claim 3, wherein the storage schema represented by Lightweight Directory Access Protocol represents a table-based system of rules organized into at least three hierarchically-organized views.

6. (Previously Presented) A client-server computer system according to claim 2, wherein said database stores validation functions stored as hierarchically-organized views that are dynamically updatable by an external administrator.

7. (Previously Presented) A client-server computer system according to claim 3, wherein said storage schema represented by Lightweight Directory Access Protocol represents validation functions stored as hierarchically-organized views that are dynamically updatable by an external administrator.

8. (Previously Presented) A client-server computer system according to claim 4, wherein said application server and said database are centrally located to said plurality of client application servers and said validation functions are maintainable by a remote administrator.

9. (Original) A client-server computer system according to claim 5, wherein said application server and said storage schema represented by Lightweight Directory Access Protocol are centrally located to said plurality of client application servers and said validation functions are maintainable by a remote administrator.

10. (Currently Amended) An application server comprising:

a plurality of client application servers, each client application server utilizing data in a particular form and each client application server generating separate requests for validation of that data wherein the requests include the data in an initial form;

means for performing validation services in response to validation requests from said plurality of client application servers, said means for performing validation services being coupled to said plurality of client application servers and the means for performing validation services performing a comparison of the data in the initial form to validation parameters stored in a hierarchical, table-based system of a plurality of rules organized into at least a FIELD view, a CLASS view and a GLOBAL view, wherein the FIELD view is the view of first priority, the CLASS view is the view of second priority and the GLOBAL view is the default view when there is no match to the data in the FIELD and CLASS views, wherein further the rules included in each of the FIELD view, CLASS view and GLOBAL view are prioritized in a predetermined execution sequence ~~a reference corresponding to the particular form used by the client application servers to determine whether the initial form matches the particular form~~ and returning to the same client application server that sent the request an indicating of valid or invalid, wherein further the data in the initial form is determined to be in a valid format when a ~~date~~ data item specified in the data in the initial form has a non-empty table field, [[a]] the ~~date~~ data item specified in the data in the initial form is in a valid ~~date~~ data format, and when the ~~date~~ data item falls within a set of range of validation values ~~dates~~; and,

means for storing and dynamically maintaining a hierarchically-organized system of validation rules coupled to said means for performing validation services wherein the validation rules are implemented by the means for performing validation in order to compare the data in the initial form to the reference.

11. (Previously Presented) An application server according to claim 10, wherein said means for storing validation rules comprises a database.

12. (Original) An application server according to claim 10, wherein said validation rules are stored in a schema in the form of Lightweight Directory Access Protocol.

13. (Previously Presented) An application server according to claim 11, wherein said database contains a table-based system of rules organized into at least three hierarchically-organized views.

14. (Original) An application server according to claim 12, wherein said schema in the form of Lightweight Directory Access Protocol represents a table-based system of rules organized into at least three hierarchically-organized views.

15. (Previously Presented) An application server according to claim 11, wherein said database stores validation functions stored as hierarchically-organized views that are dynamically updatable by an external administrator.

16. (Previously Presented) An application server according to claim 12, wherein said storage schema represented by Lightweight Directory Access Protocol represents validation functions stored as hierarchically-organized views that are dynamically updatable by an external administrator.

17. (Previously Presented) An application server according to claim 13, wherein said means for performing validation services and said database are remotely located to said plurality of client application servers and wherein said validation rules are maintainable by a remote administrator.

18. (Original) An application server according to claim 14, wherein said application server and storage schema in the form of Lightweight Directory Access Protocol are remotely located to said plurality of client application servers and further comprises means for maintaining said validation functions.

19. (Currently Amended) A system for providing an application service, the system comprising:

an application server that receives requests for data validation and that performs data validation by comparing data of the requests that is in an initial form to validation parameters stored in a hierarchical, table-based system of a plurality of rules organized into at least a FIELD view, a CLASS view and a GLOBAL view, wherein the FIELD view is the view of first priority, the CLASS view is the view of second priority and the GLOBAL view is the default view when there is no match to the data in the FIELD and CLASS views, wherein further the rules included in each of the FIELD view, CLASS view and GLOBAL view are prioritized in a predetermined execution sequence ~~a reference~~ and returns to a same application that requested the data validation an indication of valid or invalid, wherein further the data in the initial form is determined to be in a valid format when a ~~date~~ data item specified in the data in the initial form has a non-empty table field, [[a]] the ~~date~~ data item specified in the data in the initial form is in a valid ~~date~~ data format, and when the ~~date~~ data item falls within a set of range of validation values ~~dates~~;

a plurality of applications coupled to the application server, the plurality of applications utilizing the data in a particular form corresponding to the ~~reference~~ validation parameters and each application sending the data in the initial form to the application server with a separate request by each application for data validation;

one or more application programming interfaces, the one or more application programming interfaces for coupling said plurality of applications to said application server and for coupling the data validation requests and data via a plurality of computer network protocols; and

at least one dynamically-maintainable data schema coupled to said application server for providing access to data validation functions employed by the application server to compare the initial form of the data to the reference.

20. (Previously Presented) A system according to claim 19, wherein said data schema is at least partially in the form of a database.

21. (Original) A system according to claim 19, wherein said data schema comprises validation functions in the form of Lightweight Directory Access Protocol.

22. (Original) A system according to claim 20, wherein said data schema contains a table-based system of rules organized into a plurality of hierarchically-organized views.

23. (Original) A system according to claim 21, wherein said data schema in the form of Lightweight Directory Access Protocol represents a table-based system of rules organized into a plurality of hierarchically-organized views.

24. (Original) A system according to claim 20, wherein said data schema stores validation functions stored as hierarchically-organized views.

25. (Previously Presented) A system according to claim 21, wherein said data schema in the form of Lightweight Directory Access Protocol represents validation functions stored as hierarchically-organized views that are dynamically updatable.

26. (Original) A system according to claim 22, wherein said application server and said data schema are remotely located to a plurality of client application servers and said validation rules are maintainable by a remote administrator.

27. (Original) A system according to claim 23, wherein said application server and said schema in the form of Lightweight Directory Access Protocol are remotely located to a plurality of client application servers and said validation rules are maintainable by a remote administrator.

28. (Original) The system of claim 26, wherein said system couples data between the application and said application server in the form of a string.

29. (Original) The system of claim 27, wherein said application server treats data passed to it as a string.

30. (Original) The system of claim 28, wherein the said application server receives data from said application in the form of a hashtable.

31. (Currently Amended) A system for providing data validation service on requests from applications running a plurality of software protocols, the system comprising:

a data network;

an application server, the application server in communication with the data network to receive data within validation requests for validation of the initial form of the data and wherein the application server compares the initial form of the data to validation parameters stored in a hierarchical, table-based system of a plurality of rules organized into at least a FIELD view, a CLASS view and a GLOBAL view, wherein the FIELD view is the view of first priority, the CLASS view is the view of second priority and the GLOBAL view is the default view when there is no match to the data in the FIELD and CLASS views, wherein further the rules included in each of the FIELD view, CLASS view and GLOBAL view are prioritized in a predetermined execution sequence ~~a reference~~ and returns to a same application that requested the data validation an indication of valid or invalid, wherein further the data in the initial form is determined to be in a valid format when a ~~date~~ data item specified in the data in the initial form has a non-empty table field, [[a]] the ~~date~~ data item specified in the data in the initial form is in a valid ~~date~~ data format, and when the ~~date~~ data item falls within a set of range of validation values ~~dates~~;

at least one application that utilizes the data in a particular form corresponding to ~~the reference~~ the validation parameters, the application in communication with the application server, the application providing validation requests and data in the initial form to the application server via the data network and receiving the indication of valid or invalid back from the application server;

one or more open application programming interfaces, the one or more application programming interfaces capable of handling a plurality of software protocols and in communication with the application server and said applications; and

a data schema in communication with said data network, for storing validation

functions, and accessible by said application server, wherein said application server processes the validation requests and returns a response of valid or invalid to the applications according to said validation functions stored in said data schema for comparing the initial form of the data to the reference.

32. (Previously Presented) A system according to claim 31, wherein said data schema comprises a database.

33. (Original) A system according to claim 31, wherein said validation functions are stored in the format of Lightweight Directory Access Protocol.

34. (Previously Presented) A system according to claim 32, wherein said database contains a table-based system of rules organized into hierarchically-organized views.

35. (Original) A system according to claim 33, wherein said schema in the form of Lightweight Directory Access Protocol represents a table-based system of rules organized into hierarchically-organized views.

36. (Currently Amended) A system for providing an application service, the system comprising:

means for receiving a service request from a customer application computer, wherein the customer application computer requests a validation service to determine whether data in an initial form is valid or invalid, wherein further the data in the initial form is determined to be in a valid format when a ~~date~~ data item specified in the data in the initial form has a non-empty table field, [[a]] the ~~date~~ data item specified in the data in the initial form is in a valid ~~date~~ data format, and when the ~~date~~ data item falls within a set of range of validation values ~~dates~~;

means for sending a validation request instruction to an application server corresponding to data in the initial form to be validated;

means for sending a service request from the application server to a database, the service request based at least in part on the validation request;

means for performing hierarchically-based validation services on the data that is in the initial form by comparing the initial form of the data to validation parameters stored in a hierarchical, table-based system of a plurality of rules organized into at least a FIELD view, a CLASS view and a GLOBAL view, wherein the FIELD view is the view of first priority, the CLASS view is the view of second priority and the GLOBAL view is the default view when there is no match to the data in the FIELD and CLASS views, wherein further the rules included in each of the FIELD view, CLASS view and GLOBAL view are prioritized in a predetermined execution sequence ~~the reference~~;

means for remotely updating said database based on current validation requirements of said system;

means for sending a validation result to the same customer application computer that sent the request for validation and including the indication of whether the data is valid or invalid based on whether the initial form matched the validation parameters ~~reference~~ from the application server to said customer based at least in part on the validation request; and

means for providing a response to said system from said customer application computer in response to said validation result.

37. (Currently Amended) A computer-readable medium storing a plurality of instructions adapted to be executed by a processor for providing an application service, the plurality of instructions comprising instructions to:

receive a service request from a customer data device, the customer data device including data that is in an initial form to be validated by determining whether the initial form of the data matches ~~a reference~~ validation parameters stored in a hierarchical, table-based system of a plurality of rules organized into at least a FIELD view, a CLASS view and a GLOBAL view, wherein the FIELD view is the view of first priority, the CLASS view is the view of second priority and the GLOBAL view is the default view when there is no match to the data in the FIELD and CLASS views, wherein further the rules included in each of the FIELD view, CLASS view and GLOBAL view are prioritized in a predetermined execution sequence and corresponding to a particular form utilized by the customer data device;

generate a service session instruction, the service session instruction based at least in part on the service request;

send the service session instruction to one or more open application programming interfaces, the service session instruction corresponding to one or more data validation requests from said customer data device;

perform one or more validation functions based on stored rules in a database by comparing the initial form of the data to the validation parameters ~~reference~~; and

send a validation service response to the same customer data device that provided the request where the response includes the indication of valid or invalid, wherein further the data in the initial form is determined to be in a valid format when a ~~date~~ data item specified in the data in the initial form has a non-empty table field, [[a]] the ~~date~~ data item specified in the data in the initial form is in a valid ~~date~~ data format, and when the ~~date~~ data item falls within a set of range of validation values ~~dates~~, the validation service response being based on the service request.

38. (Previously Presented) A medium according to claim 37, wherein said database comprises a database and further comprises an instruction to load said database into a memory upon startup of said application service.

39. (Original) A medium according to claim 37, wherein said validation functions are stored in the format of Lightweight Directory Access Protocol and further comprise an instruction to load said database into a memory upon startup of said application service.

40. (Currently Amended) A method of providing validation data service with a client-server computer system comprising the steps of:

coupling a data validation request between a client application server and an application server, the data validation request including data in an initial form that is compared to ~~a reference~~ validation parameters that correspond[[s]] to a particular form used by the client application server, wherein the data is communicated as a set in a hashtable;

providing data validation service request instructions to a data schema in response

to said validation request coupled between said client application server and said application server;

retrieving a plurality of hierarchical dynamically maintained validation rules from a centralized storage mass coupled to said application server the validation rules for comparing the initial form of the data to the reference;

updating said validation rules;

validating data in accordance with said validation rules, wherein the validation rules are hierarchically organized into at least a first view, a second view and a third view in order of precedence and each view has its own dedicated execution sequence to process a given data field, wherein the data field is subjected to validation by the first view execution sequence, followed by the second view execution sequence, followed by the third view execution sequence; and

coupling a response that includes an indication of valid or invalid, wherein further the data in the initial form is determined to be in a valid format when a ~~date~~ data item specified in the data in the initial form has a non-empty table field, [[a]] the ~~date~~ data item specified in the data in the initial form is in a valid ~~date~~ data format, and when the ~~date~~ data item falls within a set of range of validation values ~~dates~~, wherein further the data is returned as a set in a hashtable.

41. (Currently Amended) A method for providing an application service, the method comprising:

a step for sending a data validation service request from a user computer, the data validation service request including data in an initial form;

a step for generating a validation service instruction, the service instruction based at least in part on a validation service request from said user computer;

a step for sending the service instruction to one or more data storage schemas via one or more application programming interfaces, the service instruction corresponding to one or more validation requests from the user computer;

a step for dynamically updating ~~a table of~~ validation parameters stored in a hierarchical, table-based system of a plurality of rules organized into at least a first view, a second view and a third view, wherein the first view is the view of first priority, the

second view is the view of second priority and the third view is the default view when there is no match to the data in the first and second views, wherein further the rules included in each of the first view, second view and third view are prioritized in a predetermined execution sequence where the validation rules are stored in said one or more data schemas based on changes to the application service;

a step for calling up at least one table of validation rules from said one or more data storage;

a step for performing validation functions on data in accordance with updated validation rules stored in said table and the validation request from said user computer to compare the initial form of the data to a reference corresponding to a particular form used by the user computer; and,

a step for sending a service response that includes an indication of valid or invalid, wherein further the data in the initial form is determined to be in a valid format when a ~~date~~ data item specified in the data in the initial form has a non-empty table field, [[a]] the ~~date~~ data item specified in the data in the initial form is in a valid ~~date~~ data format, and when the ~~date~~ data item falls within a set of range of validation values ~~dates~~.